Chapter 2 Getting Started with Java

Contents

Time required: 90 minutes

Read: Think Java

One of the resources we will be using in this class is:

Think Java, Second Edition
by Allen Downey and Chris Mayfield

The entire book is available in pdf: [Think Java](#)

Read the following interactive chapters.

- [Chapter 1: Computer Programming](#)

- [Chapter 2: Variables and Operators](#)

Do: Java Fundamentals Tutorials

- [Java Tutorial](#)

- [Java Intro](#)

- [Java Get Started](#) (Skip to Java Quickstart)

- [Java Syntax](#)

- [Java Output](#)

- [Java Comments](#)

- [Java Variables](#)

- [Java Data Types](#)

- [Java User Input](#)

Tutorial 1: Hello World

Hello World is the traditional first program you write in any programming language. Doing this confirms that your development environment is working.

Create a Java program named **HelloWorld.java**

```
1  // Name: HelloWorld.java
2  // Written by:
3  // Written on:
4  // Purpose: Traditional Hello World in Java
5
6  public class HelloWorld {
7      public static void main(String[] args) {
8          // Print a string literal to the console
9          System.out.println("Hello World!");
10         System.out.println("This is my first Java Application");
11     }
12 }
```

Example run:

```
Hello World!
This is my first Java Application
```

Tutorial 2: Hello Dialog

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user.

A library is a collection of prewritten code to add capabilities to a program. JOptionPane is not included in the Java standard library. This is the syntax for importing a library in Java.

```
// Import JOptionPane library for Java GUI dialog boxes
import javax.swing.JOptionPane;
```

Create a Java program named **HelloDialog.java**

```java
 1 // Name: HelloDialog.java
 2 // Written by:
 3 // Written on:
 4 // Purpose: Traditional Hello World in Java JOptionPane
 5
 6 // Import JOptionPane library for Java GUI dialog boxes
 7 import javax.swing.JOptionPane;
 8 public class HelloDialog
 9 {
10     public static void main(String[] args)
11     {
12         // Show a message to the user
13         JOptionPane.showMessageDialog(null, "Hello Java World!");
14     }
15 }
```

Example run:

Message   ×

ⓘ   Hello Java World!

OK

Tutorial 3: Miles to Kilometers

We are going to work with constants, variables, and expressions.

- **Constants:** store values that do not change

- **Variables:** store values that can change

- **Expression:** performs the work of the program

Constant: A constant is a variable whose value cannot change once it has been assigned. It is something we know before the program starts.

```java
// Constant for converting Miles to KM
final double KM_PER_MILE = 1.609;
```

Variables: A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type. A variable is the name of a reserved area

allocated in memory. In other words, it is a name of the memory location. It is a combination of "vary + able" which means its value can be changed.

```
// Variables for input and calculations
double miles = 100.0;
double kilometers;
```

Expressions: Expressions perform the work of a program. Expressions are used to compute and assign values to variables and to help control the execution flow of a program. The job of an expression is two-fold: perform the computation indicated by the elements of the expression and return some value.

```
// Convert miles to kilometers
kilometers = miles * KM_PER_MILE;
```

This program is hard coded to convert 100 miles to kilometers.

Create a Java program named: **Miles100ToKM.java**

```java
 1 // Name: Miles100ToKM.java
 2 // Written by:
 3 // Written on:
 4 // Purpose: Convert 100 miles to Kilometers
 5
 6 public class Miles100ToKM {
 7
 8     public static void main(String[] args) {
 9         // Constant for converting Miles to KM
10         final double KM_PER_MILE = 1.609;
11
12         // Variables for input and calculations
13         double miles = 100.0;
14         double kilometers;
15
16         // Convert miles to kilometers
17         kilometers = miles * KM_PER_MILE;
18
19         // Display answer to user
20         System.out.println(miles + " miles = " + kilometers + " kilometers");
21     }
22 }
```

Example run:

```
100.0 miles = 160.9 kilometers
```

Tutorial 4: Miles to Kilometers with User Input

The same program with user input using the Scanner class.

Scanner: It is used to get input from the user during runtime. The input is to be of primitive data types, like int, float, double, string, etc.

Create a Java program named: **MilesToKM.java**

```java
1  // Name: MilesToKM.java
2  // Written by:
3  // Written on:
4  // Purpose: Convert Miles to Kilometers from user input
5
6  // Import Scanner library for input
7  import java.util.Scanner;
8
9  public class MilesToKM {
10     public static void main(String[] args) {
11         // Declare Scanner object and initialize with
12         // predefined standard input object, System.in
13         Scanner keyboard = new Scanner(System.in);
14
15         // Constant for converting Miles to KM
16         final double KM_PER_MILE = 1.609;
17
18         // Variables for input and calculations
19         double miles;
20         double kilometers;
21
22         // Prompt the user for input
23         System.out.println("+------------------------------------+");
24         System.out.println("|--  Convert Miles to Kilometers  --|");
25         System.out.println("+------------------------------------+");
26         System.out.print("Enter the distance in miles: ");
27
28         // Get double from the keyboard
29         // Assign double to variable
30         miles = keyboard.nextDouble();
31
32         // Convert miles to kilometers
33         kilometers = miles * KM_PER_MILE;
34
35         // Display answer to user
36         System.out.println(miles + " miles = " + kilometers + " kilometers");
37     }
38 }
```

Example run:

```
+------------------------------------+
|--  Convert Miles to Kilometers  --|
+------------------------------------+
Enter the distance in miles: 10
10.0 miles = 16.09 kilometers
```

Assignment Submission

1. Attach the pseudocode.

2. Attach the program files.

3. Attach screenshots showing the successful operation of the program.

4. Submit in Blackboard.